Amendments to the Drawings

Please add the enclosed Formal Drawing Sheet 4/4 containing a new Figure 4.

Attachment: New Formal Drawing Sheet 4/4.

Remarks

Entrance of this Amendment, and allowance of all pending claims are respectfully requested. Upon entrance of this amendment, claims 1, 3-20 & 26-28 will remain pending, of which claims 26-28 has been withdrawn. *Should the Examiner entertain reservation regarding allowability of any claim presented herewith after entering this Amendment, the Examiner is requested to telephone Applicants' undersigned representative to discus the matter in order to facilitate prosecution of this application.*

The objections and rejections to the previously pending claims are addressed in the order raised in the Office Action of July 17, 2006.

Initially, a new Figure 4 is added herein to address the drawings objection. This new figure summarizes the substance of Applicants' specification and mirrors the language of claim 1. Specifically, the figure is added to illustrate the update stage and the separate atomic write stage of a write operation for the particular record-oriented data structure recited in the independent claims presented. The specification is amended to summarily describe new Figure 4. No new matter is added to the application by any amendment presented.

By this paper, claims 1, 8 & 20 are amended to more positively recite the steps involved in the process responsive to the 35 U.S.C. §101 rejection stated in the Office Action. Based on these amendments, withdrawal of the 35 U.S.C. §101 rejection is respectfully requested.

Further, claims 1, 8 & 20 are amended to address the 35 U.S.C. §112, second paragraph, rejection stated in the Office Action. The phrase "at least some records" is deleted from these claims, and further, the steps involved in the method are clearly set forth in each independent claim. Thus, reconsideration and withdrawal of the indefiniteness rejection to claims 1-20 is respectfully requested.

Prior claims 1-7 were rejected under 35 U.S.C. §103(a) as being unpatentable over Malcolm et al. (U.S. Patent Publication No. 2002/0004917; hereinafter Malcolm), further in view of Harris et al. (U.S. Patent No. 5,873,097; hereinafter Harris); while claims 8-20 were rejected under 35 U.S.C. §103(a) as being unpatentable of Malcolm in view of Harris, and further in view of Chan et al. (U.S. Patent No. 5,331,189; hereinafter Chan). These rejections are respectfully, but most strenuously, traversed to any extent deemed applicable to the claims presented herewith, and reconsideration thereof is requested.

Applicants recite in the independent claims presented a technique for securely managing data files in non-volatile storage. In the technique:

- The data files are stored in a record-oriented data structure with each of the records containing, in addition to data contents, a first reference indicating the current data-containing record of a previous file, and a second reference indicating the current data-containing record of a subsequent file.

In Applicants' recited record-oriented data structure, each record contains: (1) a first reference indicating the current data content record of a previous file; (2) a second reference indicating the current data-containing record of a subsequent file; *and (3) the data contents.* In accordance with Applicants' invention, the recited first and second reference are integrated with the data contents; that is, are part of the same record. For an alleged teaching of this structure, the Office Action references paragraphs [0061] & [0062] of Malcolm. These paragraphs state:

> [0061] Each of the block handles 230 includes a forward handle pointer 232, a backward handle pointer 233, a reference counter 234, a block address 235, a buffer pointer 236, and a set of flags 237.
>
> [0062] The forward handle pointer 232 and the backward handle pointer 233 reference other block handles 230 in a doubly-linked list of block handles 230.

As taught by Malcolm, each block handle 230 includes a forward handle pointer 232 and a backward handle pointer 233 which reference other block handles 230 in a doubly-linked list of block handles 230. This arrangement is illustrated in FIG. 2 of Malcolm, which also illustrates the data content, i.e., objects, such as root object 22, being separate from the binary tree 231 comprising the block handles 230. Thus, Malcolm actually teaches away from a structure such as recited by Applicants wherein each record of the record-oriented data structure includes the data contents, as well as the first reference and the second reference recited by Applicants. Malcolm does not integrate the forward handle pointer and backward handle pointer with the data content into a single record as recited by Applicants. This difference in data structures is significant.

Malcolm's invention is directed to a system for caching information objects transmitted using a computer network. A cache engine is coupled to the network and provides a cache of transmitted objects, which it stored in memory and mass storage by taking direct control of when and where to store those objects in mass memory. The caching engine determines directly when and where to store objects in memory (such as RAM) and mass storage (such as one or more disk drives) to optimally write objects to mass storage and later read them from mass storage, without having to maintain them persistently. (See paragraph [0008] of Malcolm.)

Thus, Malcolm employs both volatile and non-volatile storage. The volatile storage is labeled memory 103 in FIG. 2 and, in one example, is RAM. Because Malcolm is describing a caching function, performance is enhanced by employing the binary tree 231 in volatile memory. This is contrary to Applicants' recited invention, wherein the data files are in non-volatile storage, with the data being stored in a record-oriented data structure with each of the records containing, in addition to the data contents, the recited first reference and second reference.

In view of these above differences, Applicants respectfully submit that Malcolm does not teach or suggest Applicants' particular record-oriented data structure recited in the independent claims presented. Malcolm actually teaches away from Applicants' recited environment by teaching that the binary tree 231 comprising the block handles 230 is separate from the data contents, and further that the binary tree resides in volatile storage.

Additionally, as amended, Applicants recite performing a write operation which includes an update stage and an atomic write stage. Performing of the write operation includes performing multiple update operations for a plurality of records employing the second references (i.e., the references indicating the current data-containing record of a subsequent file) of the plurality of records. Further, the independent claims recite:

> - Each file affected by the write operation comprises a plurality of records, and for each record thereof affected by the write operation, one of the records contains existing data prior to the multiple update operations and another of the records contains corresponding data as modified according to the multiple update operations. Each of the records also includes a record-status data element indicative of the status of the data contained therein.

Thus, in Applicants' recited invention, *each record of each file affected by a write operation becomes one record which contains the data prior to the update stage and another record which contains the data subsequent to the update stage*. (Both of these records have a record-status data element indicative of the status of the data associated therewith.) This aspect of Applicants' invention was previously recited in, for example, claim 2. Regarding this aspect, the Office Action of July 17, 2006 cited (in part) column 34, lines 41-44 of Harris. This material teaches:

> Note that for updating, there can be multiple containers using the same TOC, so putting these data structure here is the most convenient way to deal with them during updating.

A careful review of the above-noted teaching fails to uncover any discussion of the particular processing recited by Applicants in the independent claims wherein *each record affected by a write operation becomes one record containing the data prior to the multiple update operations and another record containing the data as modified according the to multiple update operations (i.e., both the old and the new data are retained for each affected record).* Thus, each affected file includes the original data and the new data subsequent to the write operation(s). The above-noted teaching of Harris is believed not relevant to Applicants' recited invention. The language at issue refers to the Table of Contents (TOC) and the inclusion of the lists, that is, the set of three head/tail list pointers to doubly-link lists of the TOCObject(s). Including these lists, (i.e., data structures) in the TOC facilitates updating since there is only one TOC and one of these list sets (see column 34, lines 33-41 of Harris). A careful reading of this material fails to teach or suggest Applicants' recited characterization concerning the existence of both one record containing the data prior to the multiple update operations, and another record containing the data as modified according to the multiple update operations. Both records exist for each record of each file affected by the write operation. Column 34 of Harris does not describe keeping both records for each file affected by an update operation.

Further, Applicants recite that performing the multiple update operations includes employing the second references (each indicating the current data-containing record of a subsequent file) of the plurality of records. Column 86, lines 12-50 of Harris are cited for an alleged teaching of this concept. These lines state:

> Note, that a move only moves the value header. The value data segments "go along for the ride". By moving the value header, the user's refNum to it remains valid. The source and destination of object's touched lists are modified appropriately. The algorithm is conveniently described using a "finite state machine" (FSM). FIG. 13 is a "state-transition" diagram describing this FSM, and FIG. 13 defines the actions taken and the next state. In the table, read states on theleft, conditions across the top. The intersections are in the form of "X/s". This says "execute action X then go to state number s". The special action "NOP" means just go to the indicated state.
>
> Note, in this table, "O" and "P" stand for object and property respectively. Also, "Orig. O" and "Same O" are identical while in states 0 or 1. Here are the actions:
>
> A: if (no "from" touched list entry" create it flag "from" touched list entry as "inserted"

B: if (no "from" touched list entry" create it flag "from" touched list entry as "removed" create "to" touched list entry as "inserted" (set value header to point to this" set back pointer in "to" "inserted" entry to point to "from" "removed" entry

C: remove "inserted" flag from "from" touched list entry ("from" == "to" object) if (touched value not edited or set-infoed) delete "from" touched list entry

D: move "from" touched list ("inserted") entry to "to" object create "from" touched list entry as "removed" set back point in "to" "inserted" entry to point to "from" "removed" entry

E: move "from" touched list ("inserted") entry to "to" (original) object delete "removed" entry in "to" touched list remove "inserted" flag from "from" touched list entry if (touched value not edited or set-infoed) delete "from" touched list entry

F: move "from" touched list ("inserted") entry to "to" (original) object delete "to" "removed" touched list entry

G: move "from" touched ("inserted") list entry to "to" object State 0 occurs when an object has never moved (or was moved back to its original position).

This is the initial state. There could still be a touched list entry for the value if it was edited or set-infoed.

A careful reading of this material fails to uncover any teaching or suggestion of the relevant aspect of Applicants' recited invention. The above teachings of Harris do reference a back pointer, however, in Applicants' invention, the second reference is a forward pointer. Thus, the cited lines clearly do not teach Applicants' recited functionality. To the extent that an inherency rejection is intended, Applicants respectfully traverse such a rejection. The Office Action points to no portion of Harris to establish that their recited functionality is necessarily inherent. There is no discussion in Harris of the problem addressed by the present application, nor is there any indication that this particular functionality recited by Applicants *necessarily* flows from the teachings of Harris. Absent such a showing, it is well established that claims are to be read in their entirety, including any functional limitation presented therein.

For at least the above-noted reasons, Applicants respectfully submit that the independent claims presented herewith patentably distinguish over the purported combination of Malcolm and Harris. With respect to independent claims 8 & 20, Applicants note that Chan is further added with the combination of Malcolm and Harris for allegedly teaching Applicants' recited

invention. Chan is cited for teaching that Applicants' recited non-volatile storage in an EEPROM. Without acquiescing to this characterization of Chan, Applicants respectfully submit that a careful reading of Chan fails to teach the above-noted deficiencies of Malcolm and Harris when applied against the claims presented.

For at least the above reasons, Applicants submit that the claims presented patentably distinguish over the applied art, and request issuance of an indication of allowance thereof.

Again, should any issue remain unresolved, Applicants' undersigned representative requests a telephone interview with the Examiner to further discuss the matter in the hope of advancing prosecution of the subject application.

Respectfully submitted,

Kevin P. Radigan
Attorney for Applicants
Registration No.: 31,789

Dated: October 17, 2006.

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579